Reprinted from INFORMATION AND CONTROL
Vol. 18, No. 5, June 1971
Printed in Belgium

Cellular Automata Complexity Trade-Offs*

ALVY RAY SMITH III

Department of Electrical Engineering, New York University, New York 10453

The general theory of cellular automata is investigated with special attention to structural complexity. In particular, simulation of cellular automata by cellular automata is used to make explicit trade-off relationships between neighborhood size and state-set cardinality. A minimum neighborhood template with $d + 1$ elements is established for the class of d-dimensional cellular automata. The minimum state set for this class is shown to be the binary state set. The temporal costs, if any, of structural complexity trade-offs are also studied. It is demonstrated that any linear time cost can be eliminated and, in fact, a speed-up by arbitrary positive integer factor k can be attained at an increased structural cost.

I. INTRODUCTION

A cellular automaton—also known variously as a cellular space (von Neumann (1966), Thatcher (1964), Codd (1968)), modular computer (Wagner (1964)), iterative computer (Cole (1969)), or tessellation automaton (Moore (1962), Arbib (1966), Yamada and Amoroso (1969))—can be visualized as an infinite strip of film in one dimension or an infinite chessboard in two, each frame or square of which represents a copy of a single finite-state machine, or "cell", connected in a highly regular fashion to its neighbors. Time proceeds synchronously and in discrete steps for all machines in a cellular space. To avoid timing problems which arise when delayless communications over arbitrary distances are allowed, we assume in this paper that each cell is a Moore-type finite-state machine (i.e., unit delay between input and output).

The cellular automaton was originally employed by von Neumann in 1959 for biological modeling but has recently received renewed interest from computer designers because of technological advances. In particular, the mass fabrication techniques of large-scale integration place a premium on high

* Portions of this paper were presented at the IEEE Ninth Annual Symposium on Switching and Automata Theory, October 15–18, 1968, Schenectady, New York.

uniformity which is one of the two outstanding characteristics of the cellular automaton. The other is the high degree of parallelism inherent in the cellular model. This factor is also of interest to the computer designer in his search for faster computation times. In this paper, however, the model is treated as a mathematical object divorced from "real-world" applications. The reader interested in more specialized theory is referred to Smith (1969).

II. Definitions and Notation

Let $\mathbf{Z}$ be the integers, and let *d-D* abbreviate "*d*-dimensional" for positive integer d. Then $\mathbf{Z}^d$ is the set of d-tuples of integers and can be thought of as the points with integer coordinates in *d-D* Euclidean space E^d. In particular, the d-tuple $(0,0,\ldots,0)$ will be denoted by $\mathbf{0}$, the value of d being determined by context, and a boldface lower-case letter, say $\mathbf{a}$, will label the *d-D* vector $(a_0, a_1, \ldots, a_{d-1})$. In particular, $\mathbf{a} + \mathbf{b} = (a_0 + b_0, a_1 + b_1, \ldots, a_{d-1} + b_{d-1})$ is a vector sum. The *distance* ρ between $\mathbf{a}$ and $\mathbf{b}$ is given by $\rho(\mathbf{a}, \mathbf{b}) = |a_0 - b_0| + |a_1 - b_1| + \cdots + |a_{d-1} - b_{d-1}|$.

Then a cellular space is constructed from a countably infinite set of automata indexed by points in $\mathbf{Z}^d$ where each automaton in the set is an exact copy of a single finite-state automaton G, a *cellular-space automaton*, defined as follows:

We use the usual definition of state-output (Moore) finite automaton as a quintuple (X, Y, Q, f, β) where X, Y, and Q are finite sets called, respectively, the input states, the output states, and the internal states. The next-state function is f: $Q \times X \to Q$, and β: $Q \to Y$ gives the output. However, we will consider only the case where $X = Q^n$ and $Y = Q^n$. That is, under interpretation, G has n input lines and n output lines, which we order. That is, put $X = \prod_{i=1}^{n} Q_i$ and $Y = \prod_{j=1}^{n} Q_j$ where $Q_k = Q$, $1 \leqslant k \leqslant n$. Furthermore, we have the following restrictions on G:

(1) $\beta(q) = (q, q, \ldots, q)$;

(2) There is a specially designated state $q_0 \in Q$, the *quiescent state*, such that $f(q_0 ; q_0, \ldots, q_0) = q_0$.

For a given cellular-space automaton G and integer d, define a *d-D cell set* S to be the indexed set of exact copies of G indexed by the points in $\mathbf{Z}^d$. Thus $S = \{A_{\mathbf{m}} \mid \mathbf{m} \in \mathbf{Z}^d\}$ and each copy $A_{\mathbf{m}}$ of G is called a *cell*. A cellular space will be defined as a composition of automata in a cell set. In general, a composition is given by a composition function which specifies which inputs of automata in a set of automata are connected to which outputs of elements

of the set. Holland (1965) gives a detailed analysis of such functions. However, for our purposes the composition function can be given conveniently by specifying the neighborhood of each cell.

DEFINITION 1. Let S be a d-D cell set and let $N = (\mathbf{a}_1, \ldots, \mathbf{a}_n)$ be a finite ordered subset of $\mathbf{Z}^d$ with $\mathbf{a}_1 = \mathbf{0}$; such an N is called a *d-D neighborhood index*. Then the *neighborhood* of $A_\mathbf{m} \in S$ is the set $\{A_\mathbf{p} \mid \mathbf{p} = \mathbf{m} + \mathbf{a}_j\}$ of *neighbors* of $A_\mathbf{m}$, where input line j, $1 \leqslant j \leqslant n$, of $A_\mathbf{m}$ is output line j of $A_{\mathbf{m}+\mathbf{a}_j}$.

Thus the neighborhood of $A_\mathbf{m}$ is the set of cells which influence the state behavior of $A_\mathbf{m}$ in one time step.

DEFINITION 2. A *uniform cellular automaton* (or *uniform cellular space*) Z is a triple (G, d, N), where

- G is a cellular-space automaton with next-state function f, the *local transition function* of Z, and quiescent state q_0, the *quiescent state* of Z, and state set Q, the *state set* of Z;
- d is the *dimension* of Z;
- N is a d-D neighborhood index such that $\{A_\mathbf{p} \mid \mathbf{p} = \mathbf{m} + \mathbf{a}_j ; \mathbf{a}_j \in N\}$ is the neighborhood of $A_\mathbf{m}$ for all $A_\mathbf{m}$ in the d-D cell set S determined by G and d; N is the *neighborhood index* of Z.

Remark 1. The adjective "uniform" is used to indicate that each cell has the same neighborhood index—a restriction which might be profitably relaxed when using the model for, say, embryological modeling. We will henceforth drop the adjective, it being assumed unless otherwise stated that any cellular space discussed is a uniform cellular space.

Remark 2. Implicit in the definition is a clock which assures synchronism of operation of all cells in Z.

DEFINITION 3. The *neighborhood template* T (or just *template* T) of a cellular space Z with neighborhood index $N = (\mathbf{a}_1, \mathbf{a}_2, \ldots, \mathbf{a}_n)$ is the set $T = \{\mathbf{a}_1, \mathbf{a}_2, \ldots, \mathbf{a}_n\}$. The point $\mathbf{a}_1 = \mathbf{0}$ is the *template origin*.

For 1-D and 2-D cellular spaces it will be convenient to have a diagrammatic device for specifying a neighborhood template. Thus, e.g., a template is a subset of chessboard squares such as shown in Figure 1(a) where we hatch the cell whose neighborhood this is—i.e., we hatch the template origin. In the intuitive model, the neighborhood of cell $A_\mathbf{m}$ in space Z is determined by translating the template associated with Z until the hatched template origin

covers cell A_m; the cells that lie under the template squares then form the neighborhood of A_m.

It will often be convenient in this paper, when no confusion exists as to dimension, to label a cellular space by a pair (T, r), where T is a neighborhood template and r is the number of states per cell. This notation is used, e.g., to discuss the simulation of one space by another. In this case, the exact nature of the local transition function need not be known.

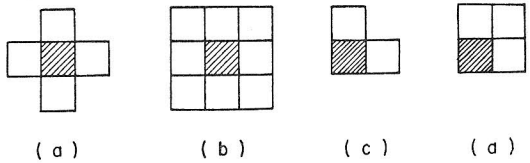

FIG. 1. Templates with hatched origins.

Two neighborhood templates of special and historical interest are the 5-cell template (Figure 1(a)) of von Neumann (1966) and the 9-cell template (Figure 1(b)) of Moore (1962). These are members of two classes of templates first defined as follows by Cole (1969):

Define two norms:

$$|\mathbf{a}| = \sum_{i=0}^{d-1} |a_i|,$$

$$\|\mathbf{a}\| = \max_{0 \leqslant i \leqslant d-1} \{|a_i|\}.$$

Then define two classes of templates from the two norms:

$$H_k = \{\mathbf{a} \mid |\mathbf{a}| \leqslant k\},$$
$$J_k = \{\mathbf{a} \mid \|\mathbf{a}\| \leqslant k\}.$$

If the general convention $T^{(d)}$ is utilized to make explicit that T is a template in a d-D cellular space, then $H_1^{(2)}$ is the von Neumann neighborhood template and $J_1^{(2)}$ is the Moore neighborhood template. We also define two closely related classes of templates:

$$K_k = \{\mathbf{a} \mid |\mathbf{a}| \leqslant k \text{ and } a_i \geqslant 0,\ 0 \leqslant i \leqslant d-1\},$$
$$L_k = \{\mathbf{a} \mid \|\mathbf{a}\| \leqslant k \text{ and } a_i \geqslant 0,\ 0 \leqslant i \leqslant d-1\}.$$

Figures 1(c) and 1(d) represent templates $K_1^{(2)}$ and $L_1^{(2)}$, respectively.

It will be convenient in discussing templates to make the following definitions: Let A and B be two subsets of $\mathbf{Z}^d$ with $\mathbf{a} \in A$ and $\mathbf{b} \in B$. Then, if $k\mathbf{a} = (ka_0, ka_1, \ldots, ka_{d-1})$ for integer k, we define the *sum* of A and B by $A + B = \{\mathbf{a} + \mathbf{b}\}$ and the *difference* of A and B by $A - B = \{\mathbf{a} + (-1)\mathbf{b}\}$. The set kA, k a nonnegative integer, is defined recursively by

$$0A = \mathbf{0},$$
$$(k+1)A = kA + A.$$

Note that $A + B$ is not the union $A \cup B$, $A \dot{-} B$ is not the set theoretic difference $A - B$, and kA is not the set $\{k\mathbf{a}\}$.

In the definitions so far, we have viewed a cellular space on the "local" level: a cellular-space automaton and a neighborhood template determine an infinite composition. We now look at the infinite composition as a single machine.

DEFINITION 4. A *configuration* c in a cellular space $Z = (G, d, N)$ with d-D cell set $\{A_\mathbf{m}\}$ and Q the internal state set of G is an assignment of states from Q to all cells in the space—i.e., c: $\mathbf{Z}^d \to Q$ is such that $c(\mathbf{m}) = q$ assigns state q to cell $A_\mathbf{m}$. The *support* of c is given by $\sup(c) = \{\mathbf{m} \mid c(\mathbf{m}) \neq q_0\}$, and $(\sup(c))' = \mathbf{Z}^d - \sup(c)$. Then C is the set of configurations in Z with finite support. A configuration in Z at time $t = 0$ is called an *initial configuration* c_0. We shall assume $\sup(c_0)$ is finite, for any c_0.

DEFINITION 5. The *neighborhood state* $\mathbf{N}(c, \mathbf{m})$ of a cell $A_\mathbf{m}$ at point $\mathbf{m} \in \mathbf{Z}^d$ in configuration c of cellular space $Z = (G, d, N)$ is defined by $\mathbf{N}(c, \mathbf{m}) = c(\{\mathbf{m}\} + N)$. Then $\mathbf{N}$ has a natural ordering induced by N.

DEFINITION 6. The *global transition function* F: $C \to C$ of cellular space Z is defined by $(F(c))(\mathbf{m}) = f(\mathbf{N}(c, \mathbf{m}))$, for all $\mathbf{m} \in \mathbf{Z}^d$. By F^t, t a positive integer, we shall mean the *t-fold composition* of F defined recursively by $F^1(c) = F(c)$ and $F^t(c) = F(F^{t-1}(c))$. Let Φ_d be the set of global transition functions for d-D cellular spaces.

Finally, we are in a position to define the principal tool used in this investigation, i.e., simulation.

DEFINITION 7. Let Z_1 and Z_2 be d-D cellular spaces with sets of configurations C_1 and C_2, respectively, and global transition functions F_1 and F_2, respectively. Let k_1 and k_2 be positive integers. Then Z_2 *simulates* Z_1 *in* k_2/k_1 *times real time* if and only if there exist effectively computable and

injective mapping $G: C_1 \to C_2$ and effectively computable function $g: \Phi_d \to \Phi_d$ such that

$$F_2^{k_2}(G(c)) = G(F_1^{k_1}(c)),$$

where $F_2 = g(F_1)$. We shall be interested in the following three cases: (1) $k_1 = 1, k_2 = k > 1$ (*k times real time*); (2) $k_1 = k > 1, k_2 = 1$ (*$1/k$ times real time*); (3) $k_1 = k_2 = 1$ (*real time*). If Z_2 simulates Z_1 in $1/k$ times real time, we shall call Z_2 a *k-speed-up* (or just *speed-up*) of Z_1.

III. Neighborhood Reduction and Speed-Up

The definition of cellular space permits neighborhoods of all sizes and shapes. Here we search for neighborhoods which contain as few cells as possible. That is, given cellular space Z, we want a cellular space Z' which simulates Z but which has a smaller neighborhood. We begin by exhibiting a theorem which states that an arbitrary neighborhood can be reduced to an H_1 neighborhood (i.e., to the von Neumann neighborhood in the 2-D case), at a cost of time and state-set size. This is of interest since much of the previous work in cellular spaces has used the von Neumann neighborhood. However, the theorem will be extended to a similar result for an even smaller neighborhood, the $K_1^{(d)}$ neighborhood, which is minimum for d-D spaces. Finally the time cost will be proved unnecessary, at the expense of a still larger state set, and, in fact, speed-ups will be demonstrated possible, even in spaces with the minimal K_1 template.

To aid in understanding the following theorem, certain convenient template nomenclature is defined and illustrated by example in Fig. 2. Let $\mathbf{v}(T)$ be the *negative extent tuple* for set $T \in \mathbb{Z}^d$ defined for $0 \leqslant i \leqslant d-1$ by

$$v_i = |\min_{t \in T} \{t_i\}|.$$

(It should be clear from context whether a pair of vertical bars indicates the absolute value of an integer, the norm of a d-tuple, or the cardinality of a set). Define $\mathbf{v}'(T)$ to be the *positive extent tuple* for set $T \in \mathbb{Z}^d$ given by

$$v_i' = \max_{t \in T} \{t_i\}.$$

Note that, if T is a template, $v_i \geqslant 0$ and $v_i' \geqslant 0$. Let $\mathbf{w}(T) = \mathbf{v}(T) + \mathbf{v}'(T)$ denote the *total extent tuple* for set $T \in \mathbb{Z}^d$. We shall refer to $\mathbf{v}$, $\mathbf{v}'$, and $\mathbf{w}$

as *extent tuples*. Finally, from the extent tuples $\mathbf{v}$ and $\mathbf{v}'$, define the *minimal prism* $P(T)$ for set $T \in \mathbb{Z}^d$ by $P = \{\mathbf{a} \mid -v_i \leqslant a_i \leqslant v_i'\}$. The dashed line in Fig. 2 indicates the minimal prism for the template shown.

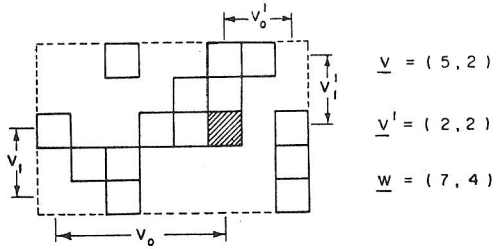

FIG. 2. Template nomenclature.

THEOREM 3.1. *For an arbitrary d-D* (T, r) *cellular space* Z, *there exists a d-D* (H_1, s) *cellular space* Z' *which simulates* Z *in* m *times real time, where, for extent tuples* $\mathbf{v}(T)$, $\mathbf{v}'(T)$, *and* $\mathbf{w}(T)$,

$$m = \sum_{j=0}^{d-1} \max(v_j, v_j'),$$

and $s = s_{d-1}(s_{d-2} + 1)^{-2}$ *with* s_i *defined recursively for* $-1 \leqslant i \leqslant d-1$ *by*

$$\begin{aligned} s_{-1} &= r, \\ s_0 &= r(r+1)^{w_{d-1}}, \\ s_i &= s_{i-1}(s_{i-1}+1)^{w_{d-(i+1)}}. \end{aligned}$$

Proof. Successive pooling of the information in each dimension is the idea behind this proof. We prove the theorem for only the 2-*D* case here and leave the straightforward induction to the *d-D* case for the interested reader. Let Z have transition function f and state set Q. The proof is in two steps, one for reduction in each dimension.

Step 1. Reduce Z to $(T_1, r(r+1)^{w_1})$ cellular space Z_1, where T_1 is as illustrated in Fig. 3, by supplying each cell in Z_1 with states of the set $Q_1' = Q_1^{v_1'} \times Q \times Q_1^{v_1}$, where $Q_1 = Q \cup \{b\}$ and b is a specially designated state. For each cell in Z in state $q \in Q$, there is a corresponding cell in Z_1 set to state $q_{11}' = (b, \ldots, b, q, b, \ldots, b)$. Z_1 is supplied with a transition function f_1 which "fills in the blanks" b of q_{11}' as follows:

Number the positions of the $(w_1 + 1)$-tuple q'_{11} from 0 to w_1, from left to right (then q is in position v_1'). Let cell D be in state q'_{11}. Then f_1 changes q'_{11} to $q'_{12} = (b,..., b, \overline{v_1'}, q, \underline{v_1'}, b,..., b)$ where $\overline{v_1'}$ is the value of position v_1' in the cell above (i.e., the positive 1 direction) cell D and $\underline{v_1'}$ is the value of position v_1' in the cell below D. Similarly, f_1 changes q'_{12} to

$$q'_{13} = (b,..., b, \overline{v_1' - 1}, \overline{v_1'}, q, \underline{v_1'}, \underline{v_1' + 1}, b,..., b),$$

and so forth until q'_{1y} contains no symbols b, where $y = \max(v_1, v_1')$. Note that this process requires $\max(v_1, v_1')$ steps.

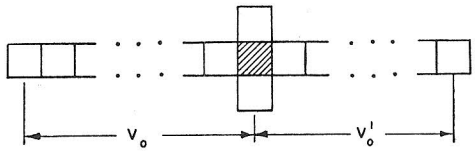

FIG. 3. Template T_1.

Then q'_{1y} contains the value of position v_1' of every cell up to and including the cell v_1' cells above D and of every cell up to and including the cell v_1 cells below D. That is, the neighborhood state of a cell in Z_1 in state q'_{1y} contains at least the information contained in a neighborhood state of a cell in state q in Z. Transition function f_1 selects this information, acts on it as would f, and resets the b's all in one additional step. Thus Z_1 simulates Z with the given number of states.

Step 2. Reduce Z_1 to $(H_1, s_0(s_0 + 1)^{w_0-2})$ cellular space Z_2 where $s_0 = r(r + 1)^{w_1}$. Then identify Z' with Z_2. This can be accomplished in a manner analogous to that used for reducing Z to Z_1 in Step 1. Each state in Z_2 is of the set $Q_2' = Q_2^{v_0'-1} \times Q' \times Q_2^{v_0-1}$, where $Q_2 = Q_1' \cup \{B\}$, with $B \neq b$ a specially designated state. For each cell in Z_1 in state $q'_{1y} \in Q_1'$ there is a cell in Z_2 in state $q'_{21} = (B,..., B, q'_{1y}, B,..., B) \in Q_2'$ and f_2 fills in the blanks B, just as did f_1 for Z_1, to obtain q'_{2z}, where $z = \max(v_0 - 1, v_0' - 1)$. The Z_2 neighborhood thus contains the information in a Z_1 neighborhood, and f_2 is designed to act on this information to simulate Z_1 and hence Z. The blank-filling process requires z steps and one step is needed to reset the B's.

Clearly, Step 2 can follow Step 1 immediately if the final reset operation of Step 1 is omitted. Then filling all blanks b and B requires $\max(v_1, v_1') + \max(v_0 - 1, v_0' - 1)$ steps and resetting them requires one

step. The simulation of one step of f occurs during the reset step. Hence Z is simulated in m times real time where

$$m = 1 + \max(v_1, v_1') + \max(v_0 - 1, v_0' - 1) = \max(v_0, v_0') + \max(v_1, v_1').$$

Q.E.D.

COROLLARY 3.1.1. *For an arbitrary d-D (T, r) cellular space Z, there exists a d-D (H_1, s) cellular space Z' which simulates Z in m times real time, where $m = d\alpha$, for*

$$\alpha = \max_{t \in T} \| t \|,$$

and $s < (r + d)^n$, for $n = (2\alpha + 1)^d = | J_\alpha^{(d)} |$.

Proof. For s_i defined as in Theorem 3.1, $s_i = s_{i-1}(s_{i-1} + 1)^{w_{d-(i+1)}} \leqslant s_{i-1}(s_{i-1} + 1)^{2\alpha} < (s_{i-1} + 1)^{2\alpha+1}$. In particular, $s_0 < (r + 1)^{2\alpha+1}$ and $s_1 < (s_0 + 1)^{2\alpha+1}$. But, for positive integers x and a, $(x^a + 1) \leqslant (x + 1)^a$. So $s_1 < (r + 2)^{(2\alpha+1)^2}$. Induction on the index i of s_i gives $s_i < (r + i + 1)^{(2\alpha+1)^{i+1}}$. The corollary follows from $s = s_{d-1}(s_{d-2} + 1)^{-2} < s_{d-1}$. Q.E.D.

Another corollary to Theorem 3.1 is immediate from the method of proof there: For every i for which $v_i' = 0$, one cell can be removed from the H_1 template in the theorem. In particular, the cell at $(0, 0, \ldots, i, \ldots, 0)$ can be removed. Similarly, for every i for which $v_i = 0$, the cell at $(0, 0, \ldots, -i, \ldots, 0)$ can be removed. Thus, if the origin of template T falls on a face of its minimal prism, then no information-gathering "arm" is required in the simulating H_1 template in the direction perpendicular to this face. One might be led from such considerations to the belief that the position of the origin in a template indicates the possible directions of information flow in a cellular space. For example, the $K_1^{(1)}$ template $\{0, 1\}$ seems to imply that information can be transferred from right to left only—i.e., in general, the minimal 1-D template is the $H_1^{(1)}$ template. But these ideas are scotched in the next theorems.

THEOREM 3.2. *There is a (K_1, s) cellular space Z' which simulates a given (H_1, r) cellular space Z in $(d + 1)$ times real time, where Z and Z' are d-D and $s = r(r + 1)^{d+1}$.*

Proof. As in the proof of Theorem 3.1, each cell of Z' has a state set partitioned into enough coordinates to contain sufficient information about an H_1 neighborhood to simulate Z. Note that by passing information via state coordinates, a K_1 neighborhood can see in m time steps what a K_m

neighborhood sees in one step. We only require that m be large enough for K_m to contain a set of cells of the shape H_1.

Consider the set $\mathbf{H}_1 = \{\mathbf{h} + \mathbf{e} \mid \mid \mathbf{h} \mid \leqslant 1\}$, where $\mathbf{e} = (1, 1, \ldots, 1)$. Thus, $\mathbf{H}_1 \subseteq \mathbf{H}_1'' = \{\mathbf{h}' \mid d - 1 \leqslant \mid \mathbf{h}' \mid \leqslant d + 1; h_i' \geqslant 0, 0 \leqslant i < d\}$. Therefore, in $d + 1$ steps $(d + 1)\, K_1 = K_{d+1} \supseteq \mathbf{H}_1$. In d steps $dK_1 = K_d$ includes the set $\mathbf{H}_1' = \{\mathbf{h}' \mid d - 1 \leqslant \mid \mathbf{h}' \mid \leqslant d, h_i' \geqslant 0\} \cap \mathbf{H}_1 \not\supseteq \mathbf{H}_1$. Each state in Z' needs a coordinate for one state of Z and a coordinate for each element of $\mathbf{H}_1'$. Utilizing a blank symbol as before, we thus require $s = r(r + 1)^{|\mathbf{H}_1'|}$. But $| \mathbf{H}_1' | = | H_1 | -$ (the number of points in H_1 with coordinates of form $(0, 0, \ldots, 1, \ldots, 0)) = | H_1 | - d = (2d + 1) - d = d + 1$. Q.E.D.

THEOREM 3.3. *There is an (L_1, s) cellular space Z' which simulates a given (J_1, r) cellular space Z in 2 times real time, where Z and Z' are $d - D$ and $s = r(r + 1)^{2^d - 1}$.*

Proof. We proceed as in the proof above. Thus for $\mathbf{e} = (1, 1, \ldots, 1)$, $\mathbf{J}_1 = \{\mathbf{a} + \mathbf{e} \mid \| \mathbf{a} \| \leqslant 1\} = \{\mathbf{a}' \mid 0 \leqslant \| \mathbf{a}' \| \leqslant 2, a_i' \geqslant 0, 0 \leqslant i < d\}$. Therefore in two steps, $2L_1 = L_2 = \mathbf{J}_1$, and in one step L_1 includes $\mathbf{J}_1' = \{\mathbf{a}' \mid 0 \leqslant \| \mathbf{a}' \| \leqslant 1, a_i' \geqslant 0\} \not\supseteq \mathbf{J}_1$. In fact, $\mathbf{J}_1' = L_1$ and hence includes point $\mathbf{0}$. Since an extra coordinate is not required for $\mathbf{0}$, then $s = r(r + 1)^{|L_1| - 1}$. But $| L_1 | = 2^d$. Q.E.D.

Remark 1. Both simulations above "slide" in time. That is, the simulated configuration appears to move with respect to the origin of Z' whereas it is stationary with respect to the origin of Z. However, the amount of slip is effectively determined. It is interesting to notice that, if $\mathbf{0}$ is *not* required to be a template element, then the necessary template size for simulation of arbitrary $d - D$ cellular spaces can be reduced even further to d. We let the reader check that this is true by imitation of the simulations above.

Remark 2. Clearly there are numerous isomorphisms of the two simulations above obtained simply by "rotations" of the L_1 or K_1 templates. In fact, there are 2^d such straightforward isomorphisms in each case.

Apply Theorem 3.2 to Corollary 3.1.1 to obtain the next result.

THEOREM 3.4. *For an arbitrary $d - D$ (T, r) cellular space Z, there exists a $d - D$ (K_1, s) cellular space Z' which simulates Z in m times real time where $m = d(d + 1)\alpha$, for $\alpha = \max_{\mathbf{t} \in T} \| \mathbf{t} \|$ and $s < ((r + d)^n + 1)^{d+2}$, for $n = (2\alpha + 1)^d$.*

Note that the cost of a reduced neighborhood in the simulations thus far introduced is not only an enlarged state set but also a time slow-down.

In the simulations below we show how to reduce the number of neighbors at a cost of increased state set cardinality only. This is obtained by a many-one correspondence between simulated and simulating cellular spaces. That is, a set of cells in Z will be mapped into a single cell of Z'. Not only can real time be maintained with reduced neighborhood but speed-ups can be obtained.

We will need the following lemma which is adapted from Cole (1969). The idea is to use a homomorphism h to "spread out" a desired template T' of a simulating space Z'. Then one can think of overlaying cells in a space Z with the spread-out template $h(T') = \{h(\mathbf{t}') \mid \mathbf{t}' \in T'\}$. The problem is to find a set K of points with which to augment each point of $h(T')$ so that the following holds: The set of cells $h(\mathbf{t}') + K$, treated as one cell with template $h(T') + K$, has access to as much information in one time step as has the set of cells $h(\mathbf{t}') + K$ in k time steps with template T of the simulated space Z. The following condition on h and K is sufficient in the sense that if it is satisfied then it is a straightforward matter to design an appropriate transition function for the simulating space Z'.

LEMMA 3.5. *Let (T, r) and (T', r') be $d - D$ cellular spaces Z and Z', respectively. Let h be an injective homomorphism from the additive group $\mathbf{Z}^d$ into $\mathbf{Z}^d$, and let $K \subseteq \mathbf{Z}^d$ be a finite set of points. Define the state set of a cell in Z' at point $\mathbf{x}$ to be the Cartesian product of the state sets of the cells at points in $\{h(\mathbf{x})\} + K$ in Z. Then a sufficient condition that a transition function exists for simulation of Z by Z' in $1/k$ times real time is that*

$$h(T') + K \supseteq kT + K.$$

Proof. Let $Z_{\mathbf{p}}$ be the cell in Z at $\mathbf{p}$. Clearly, the state of $Z_{\mathbf{p}}$ at time $t + k$ is completely determined by the states of $Z_{\mathbf{q}}$ such that $\mathbf{q} \in \{\mathbf{p}\} + kT$. In particular, the state of each component $Z_{h(\mathbf{x})+\mathbf{a}}$ of $Z_{\mathbf{x}}'$, $\mathbf{a} \in K$, is completely determined in k steps by the states of $Z_{\mathbf{q}}$, $\mathbf{q} \in \{h(\mathbf{x})\} + \{\mathbf{a}\} + kT$. Thus in k steps, the next state of $Z_{\mathbf{x}}'$ is determined by the cells in Z at $\{h(\mathbf{x})\} + K + kT$.

But the next state of $Z_{\mathbf{x}}'$ is determined by the states of cells in Z' at $\{\mathbf{x}\} + T'$. Each one of these cells is the Cartesian product of cells in Z at $\{h(\mathbf{x} + \mathbf{t}')\} + K$. Hence $Z_{\mathbf{x}}'$ can be computed if $h(\{\mathbf{x}\} + T') + K \supseteq \{h(\mathbf{x})\} + K + kT$. Since h is a homomorphism, $h(\{\mathbf{x}\} + T') = \{h(\mathbf{x})\} + h(T')$ and the condition becomes $h(T') + K \supseteq K + kT$. Q.E.D.

THEOREM 3.6. ***For an arbitrary (T, r) cellular space Z, there is a (J_1, s) cellular space Z' which simulates Z in $1/k$ times real time, where Z and Z' are $d - D$ and $s = r^{|P(kT)|}$, for k an arbitrary positive integer.***

Proof. Define h and K as done below to satisfy the condition of Lemma 3.5. Then a cell in Z' at $\mathbf{x}$ will be the "macrocell" of cells in Z at $\{h(\mathbf{x})\} + K$. In particular, form kT and let $K = P(kT)$. Then h is defined as follows:

$$h((x_0, x_1, \ldots, x_{d-1})) = (u_0x_1, u_1x_1, \ldots, u_{d-1}x_{d-1}),$$

where $u_i = \max(v_i, v_i')$ for extent tuples $\mathbf{v}$ and $\mathbf{v}'$ of kT. These definitions for h and K ensure that $h(J_1) + K \supseteq kT + K$. This is so because $h(J_1) + K = \{\mathbf{a} \mid -2v_i \leqslant a_i \leqslant v_i + v_i' \text{ if } v_i \geqslant v_i' \text{ and } -v_i - v_i' \leqslant a_i \leqslant 2v_i' \text{ else}\} \supseteq \{\mathbf{a} \mid -2v_i \leqslant a_i \leqslant 2v_i'\}$. But $P(kT) \supseteq kT$; hence $2P \supseteq kT + K$ where $2P = \{\mathbf{b} \mid -2v_i \leqslant b_i \leqslant 2v_i'\}$. Thus, by Lemma 3.5, there exists a transition function such that Z' simulates Z with a k-speed-up, and $s = r^{|K|}$, where $|K| = |P|$. Q.E.D.

COROLLARY 3.6.1. *For any (T, r) $d - D$ cellular space Z there is an (L_1, s) $d - D$ cellular space Z which simulates Z in $1/k$ times real time, for arbitrary positive integer k.*

Proof. Use Theorem 3.6 to obtain space Z'' which simulates Z in $1/2k$ times real time. Then apply Theorem 3.3 to Z''. Q.E.D.

It is possible, but surprisingly difficult, to reduce a cellular space with the Moore template to a cellular space with the von Neumann template. Cole (1969) has shown that $h(\mathbf{x}) = (2d^2 + 1)\mathbf{x}$ and $K = H_{d^2} + J_{d^2}$ satisfy the condition of Lemma 3.5 for $k = 1$ and dimension d. Hence there is an (H_1, r^n) cellular space which simulates a given (J_1, r) cellular space in real time, with $n = |H_{d^2} + J_{d^2}|$. This result is improved below, after introduction of a new class of templates.

Let the *length* of $\mathbf{x} \in \mathbf{Z}^d$ be the positive real number

$$l(\mathbf{x}) = (x_0^2 + x_1^2 + \cdots + x_{d-1}^2)^{1/2}.$$

Then S_a is the class of templates defined by

$$S_a^{(d)} = \{\mathbf{x} \mid l(\mathbf{x}) \leqslant a\}$$

for $a > 0$ a real number.

THEOREM 3.7. *For an arbitrary $d - D$ (J_1, r) cellular space Z, there exists a $d - D$ (H_1, r^n) cellular space Z' which simulates Z in real time, with $n = |S_a^{(d)}|$ and $a = 2\lfloor d^{3/2} \rfloor$.*

Proof. Define $h(\mathbf{x}) = c\mathbf{x}$, $c = 2d$, and $K = S_a$, $a = 2\lfloor d^{3/2} \rfloor$. Since $S_a + J_1 \subseteq S_{a+d^{1/2}}$, we shall demonstrate that $S_{a+d^{1/2}} \subseteq h(H_1) + S_a$ and, hence, that h and K satisfy Lemma 3.5 for $k = 1$.

Let $l \equiv l(\mathbf{x})$. Then for $l \leqslant a$, $\mathbf{x} \in S_a \subseteq h(H_1) + S_a$. So assume $l > a$. We wish to find d-tuples $\mathbf{b} \in S_a$ and $\mathbf{e} \in h(H_1)$ such that $\mathbf{x} = \mathbf{b} + \mathbf{e}$. Let x_μ be an element of $\mathbf{x}$ such that $|x_\mu| \geqslant x_i$, $0 \leqslant i \leqslant d-1$. Without loss of generality, take $\mu = 0$ and $x_\mu > 0$. Then define $\mathbf{e} = (c, 0, 0, \ldots, 0)$ and $\mathbf{b} = (x_0 - c, x_1, x_2, \ldots, x_{d-1})$. Clearly, $\mathbf{e} \in h(H_1)$. Hence we must show $\mathbf{b} \in S_a$ or, since $(l(\mathbf{b}))^2 = l^2 - 2cx_0 + c^2$, that

$$l^2 - 2cx_0 + c^2 \leqslant a^2 \tag{1}$$

for all $\mathbf{x}$ such that $a < l \leqslant a + d^{1/2}$.

For a given length l the minimum value for x_0 is $\min x_0 = \lceil l/d^{1/2} \rceil$. Define $\delta(l) = l^2 - 2cl/d^{1/2} + (c^2 - a^2)$, for parameter d. If $\delta(l) \leqslant 0$, then (1) is valid for $x_0 = \min x_0$, since $\min x_0 \geqslant l/d^{1/2}$. If (1) is valid for $\min x_0$, then it is true for all $\mathbf{x}$ of the same length l but larger x_0. Hence it suffices to prove $\delta(l) \leqslant 0$, $a < l \leqslant a + d^{1/2}$.

Note that $\delta(l)$ is an upward opening parabola, with a vertex at $l = cd^{-1/2}$, $\delta = c^2(d-1)/d - a^2$. We shall want $\delta(l) = 0$ for some l hence require $a^2 \geqslant c^2(d-1)/d$. Let l_0 be the length such that $\delta(l_0) = 0$. Then $\delta(l) \leqslant 0$ if $0 < l \leqslant l_0$. Let $l_0 \geqslant a + d^{1/2}$. Then $\delta(a + d^{1/2}) \leqslant 0$ implies $(a + d^{1/2})^2 - 2cd^{-1/2}(a + d^{1/2}) + (c^2 - a^2) \leqslant 0$. With $c = 2d$, solve for a to get the condition $a \geqslant 2d^{3/2} - 3d^{1/2}/2$. The choice $a = 2\lfloor d^{3/2} \rfloor$ satisfies this condition and also the condition that $a^2 \geqslant c^2(d-1)/d$. Q.E.D.

COROLLARY 3.7.1. *For any $d - D$ (T, r) cellular space Z there exists a $d - D$ (H_1, s) cellular space Z' which simulates Z in $1/k$ times real time, for arbitrary positive integer k.*

Take Corollary 3.7.1 with a $k(d+1)$-speed-up and apply Theorem 3.2 to obtain the concluding result of this section.

COROLLARY 3.7.2. *For any $d - D$ (T, r) cellular space Z there exists a $d - D$ (K_1, s) cellular space Z' which simulates Z in $1/k$ times real time, for arbitrary positive integer k.*

IV. State-Set Reduction

In the preceding section we substituted large cells—i.e., cells with a large number of states—with small neighborhoods for small cells with large

neighborhoods. Now we show how to substitute small cells with large neighborhoods for large cells with small neighborhoods. In particular, it will be shown that any cellular space with the von Neumann neighborhood can be reduced to a binary cellular space (i.e., with only two states per cell). The result also holds for cellular spaces with arbitrary neighborhood because of Corollary 3.7.1. We assume the H_1 template only for ease of presentation.

The proof of the main result of this section, Theorem 4.2, requires a special type of number M for uniquely specifying position information in a simulating cellular space. Assume it is binary and of length n. Then we require M to be such that, if its bits are evenly spaced around a circle, each substring of m successive bits is distinct. For example, the number 00111 is such a number. The important point is that each position i, $0 \leqslant i \leqslant n-1$ of M is uniquely coded by the substring at positions $\{(i+j) \bmod n \mid 0 \leqslant j < m-1\}$ if $n \leqslant 2^m$. Such a number is called a *binary shift-register sequence of degree m* (see, e.g., Golomb (1967)). It is said to be *zero-free* if it does not contain the substring 0^m (i.e., a substring of m copies of symbol 0).

LEMMA 4.1 (Golomb). *There is a binary zero-free shift-register sequence of degree m and length n, for any m and n such that* $n < 2^m$.

Note that by juxtaposing copies of a shift-register sequence, one does not have to return to the beginning of a sequence to complete a cycle. The necessary bits are the initial bits of the succeeding sequence. This is the mode of "cycling" used in the following proof.

THEOREM 4.2. *For an arbitrary* $d-D$ (H_1, r) *cellular space Z, there exists a* $d-D$ $(T', 2)$ *cellular space Z′ which simulates Z in real time, where* $|T'| = 4nd - 2d + m - 1$ *for n the minimum integer such that* $r \leqslant 2^n - n$ *and m the minimum integer such that* $n < 2^m$.

Proof. Solve for m and n in the inequalities of the theorem. We shall justify these inequalities below. Then Z' is designed to have a distinct $1-D$ array of cells of length $2n$ for each cell in Z. The states of n of these cells (every other one) represent a binary code for the state of the cell in Z they simulate. The other n cells provide position information to the transition function f' of Z'.

More specifically, let $a/\mathbf{x}$ denote the point $(a, x_1, x_2, \ldots, x_{d-1})$, where $\mathbf{x} = (x_0, x_1, x_2, \ldots, x_{d-1})$ and a is an integer; then $x_0/\mathbf{x} = \mathbf{x}$. Let c and c' be configurations in Z and Z', respectively. Then string

$$\sigma_1 = c'(2nm_0/\mathbf{m})\, c'(2nm_0 + 2)/\mathbf{m}) \cdots c'((2n(m_0 + 1) - 2)/\mathbf{m})$$

is a binary "state code" for $c(m_0/\mathbf{m})$. The cells of Z' in the set $\sum_1(\mathbf{m}) = \{2nm_0/\mathbf{m}, (2nm_0 + 2)/\mathbf{m}, \ldots, (2n(m_0 + 1) - 2)/\mathbf{m}\}$ are called "state-code cells", for any $\mathbf{m}$ in Z. Z' has two states, 0 and 1; assume 0 is the quiescent state of Z'. Then $\sigma_1 = 0^n$ is the code for the quiescent state of Z.

Let string

$$\sigma_2 = c'((2nm_0 + 1)/\mathbf{m})\, c'((2nm_0 + 3)/\mathbf{m}) \cdots c'((2n(m_0 + 1) - 1)/\mathbf{m})$$

be called the "position code" of Z'. The same position code is used for each $\mathbf{m}$ in Z. All cells which are not state-code cells are thus "position-code cells". The position code is taken to be a binary zero-free shift-register sequence of degree m and length n. Lemma 4.1 guarantees the existence of such a sequence for $n < 2^m$. We insist on zero-free sequences to avoid the confusion between 0^m as part of a position code and as part of $(\sup(c_0'))'$, where c_0' is the initial configuration of Z'. There are n distinct cyclic permutations of the position code of Z'. We insist that these n codes not be used as state codes. Hence the condition $r + n \leqslant 2^n$. This plus the requirement that 0^n be the quiescent state code are the only conditions on the choice of state codes.

Note that the position of a state-code cell in a state code is uniquely specified by the states of m position-code cells. In particular, let the cell at $\mathbf{x} \in \sup(c')$ be a state-code cell in position i of a state code, $0 \leqslant i \leqslant n - 1$. Then the cells in set $\Sigma_2(\mathbf{x}) = \{(x_0 + 1)/\mathbf{x}, (x_0 + 3)/\mathbf{x}, \ldots, (x_0 + 2m - 1)/\mathbf{x}\}$ are position-code cells and the string

$$c'((x_0 + 1)/\mathbf{x})\, c'((x_0 + 3)/\mathbf{x}) \cdots c'((x_0 + 2m - 1)/\mathbf{x})$$

is a unique code for position i.

Give Z' the template $T' = ((\gamma(H_1) + \Sigma_1(\mathbf{0})) \dot{-} \Sigma_1(\mathbf{0})) \cup \Sigma_2(\mathbf{0})$, where γ maps any point $\mathbf{m}$ in Z into point $2nm_0/\mathbf{m}$ in Z'. Thus $|T'| = n|H_1| + (2d - 1)(n - 1) + m = 4nd - 2d + m - 1$. If a cell is a state-code cell, then T' supplies sufficient information to f' for simulation of Z. If a cell is a position-code cell in $\sup(c')$, then f' does not change its state. The two types of cells can be distinguished by f' from the information provided by template elements $\Sigma_1(\mathbf{0})$: If the string of state symbols from these cells is a state code, then the cell is a state-code cell. If the string is a cyclic permutation of the position code, then the cell is a position-code cell.

Let $\sup \#(c_0)$ be the support of the initial configuration of Z augmented by a boundary of quiescent cells—i.e., $\sup \#(c_0) = \sup(c_0) + H_1^{(d)}$. Then the initial configuration c_0' in Z' consists of the state codes and position codes

for all cells in sup $\#(c_0)$ and is everywhere else quiescent. Since $|\sup(F(c))|$ may, in general, be larger than $|\sup(c)|$ in Z, the position code in Z' must be propagated throughout the space. But the template T' supplies sufficient information for f' to spread the position code:

Consider a cell $A_{\mathbf{m}}$ at $\mathbf{m}$ in Z'. If cells at $\{\mathbf{m}\} + \Sigma_2(\mathbf{0})$ are quiescent, then $A_{\mathbf{m}}$ is not a state-code cell in $\sup(c')$. If, in addition, the cells at $(\{\mathbf{m}\} + \Sigma_1(\mathbf{0})) \dot{-} \Sigma_1(\mathbf{0})$ contain nonquiescent states but not the position code, then $A_{\mathbf{m}}$ is a position-code cell not in $\sup(c')$ and hence not yet coded as part of the position code. If, however, $\{\mathbf{m}\} + T'$ contains the position code, then f' codes $A_{\mathbf{m}}$ depending on the placement of the position code in its neighborhood. We complete the proof by sketching one possible scheme for this calculation:

If $(\{\mathbf{m}\} + \Sigma_1(\mathbf{x})) \dot{-} \Sigma_1(\mathbf{x})$, for $\mathbf{x} \in H_1 - \{\mathbf{0}, -1/\mathbf{0}, 1/\mathbf{0}\})$, contains the position code then the next state of $A_{\mathbf{m}}$ is simply the present state of $A_{\mathbf{m}+\mathbf{x}}$. If, for $\mathbf{x} \in \{-1/\mathbf{0}, 1/\mathbf{0}\}$, $(\{\mathbf{m}\} + \Sigma_1(\mathbf{x})) \dot{-} \Sigma_1(\mathbf{x})$ contains the position code, then the next state of $A_{\mathbf{m}}$ is the present state of $A_{\mathbf{m}+2nx_0/\mathbf{x}}$. Q.E.D.

V. Summary

This paper is an initial study of cellular automata on a general mathematical level—in particular, only the number of states per cell and the neighborhood template of any of the cellular automata studied are assumed given. Let p be the number of cells in a template and let q be the number of states per cell. Then the complexity measure pq has been the object of study here.

The K_1 neighborhood templates have been shown to be minimal, in general. Thus, in the class of d-dimensional cellular automata, there is always a cellular automaton with complexity product $(d + 1)\, q'$ which simulates an arbitrary cellular automaton with product pq, where $q' > q$ and p is arbitrary. On the other hand, in the class of d-dimensional cellular automata, we have shown that there is always a cellular automaton with complexity measure $2p'$ which simulates an arbitrary cellular automaton with product pq, with $p' > p$ and q arbitrary.

Finally, we have proved the existence of k-speed-ups. That is, in the class of d-dimensional cellular automaton, there is always a cellular automaton with complexity $p'q'$ which simulates in $1/k$ times real time an arbitrary cellular automaton with complexity pq, with $q' > q, p' \leqslant p$, and k an arbitrary positive integer.

Special applications of the general theory herein lead to quite large

reductions in the complexity product. The specialization to cellular automata which compute partial recursive functions is a good example and is treated in detail in Smith (1969).

ACKNOWLEDGMENT

I thank Michael Arbib for his assistance and guidance and dedicate Theorem 3.7 to Andrea Kaufman.

RECEIVED: December 23, 1969; REVISED: September 23, 1970

REFERENCES

ARBIB, M. A. (1966), A simple self-reproducing universal automaton, *Information and Control* **9**, 177–189.

CODD, E. F. (1968), "Cellular Automata," Academic Press, New York.

COLE, S. N. (1969), Real-time computation by n-dimensional iterative arrays of finite-state machines, *IEEE Trans. Comput.* **C-18**, 349–365.

GOLOMB, S. W. (1967), "Shift-Register Sequences," Holden-Day, San Francisco.

HOLLAND, J. H. (1965), Universal embedding spaces for automata, *in* "Progress in Brain Research," Vol. 17 (Cybernetics of the Nervous System) (N. Wiener and J. P. Schade, Eds.), pp. 223–243, Elsevier, New York.

MOORE, E. F. (1962), Machine models of self-reproduction, *in* "Proceedings of Symposia in Applied Mathematics," Vol. XIV, pp. 17–34, American Mathematical Society, Providence, R. I.

SMITH, A. R., III (1969), "Cellular Automata Theory," Technical Report No. 2, Digital Systems Laboratory, Stanford University, Stanford, Calif.

THATCHER, J. W. (1964), "Universality in the von Neumann Cellular Model," Technical Report 03105-30-T, ORA, The University of Michigan, Ann Arbor, Michigan.

VON NEUMANN, J. (1966), "The Theory of Self-Reproducing Automata," (A. W. Burks, Ed.), University of Illinois Press, Urbana, Illinois.

WAGNER, E. G. (1964), "An Approach to Modular Computers: I. Spider Automata and Embedded Automata," IBM Research Report, IBM RC 1107, IBM T. J. Watson Research Center, Yorktown Heights, New York.

YAMADA, H. AND AMOROSO, S. (1969), Tessellation automata, *Information and Control* **14**, 299–317.

Printed in Belgium by the St. Catherine Press Ltd., Tempelhof 37, Bruges